



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/663,563	09/15/2000	Thomas E. Saulpaugh	5181-72000	3700
7590 Robert C Kowert conley Rose & Tayon PC PO Box 398 Austin, TX 78767-0398		08/02/2007	EXAMINER KANG, INSUN	
			ART UNIT 2193	PAPER NUMBER
			MAIL DATE 08/02/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

09/663,563

Applicant(s)

SAULPAUGH ET AL.

Examiner

Insun Kang

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 April 2007.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,2,4-41,43-71 and 73-90 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1, 2, 4-41, 43-71, and 73-90 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the amendment filed on 4/23/2007.
2. As per applicant's request, claims 3, 42, and 72 have been canceled, claims 1, 4, 40, 43, 71, and 73 have been amended. Claims 1, 2, 4-41, 43-71, and 73-90 are pending in the application.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1, 2, 4-41, 43-71, and 73-90 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-66 of U.S. Patent No. 7,200,848 ('848).

Although the conflicting claims are not identical, they are not patentably distinct from each other because they are directed to substantially the same invention and recites only obvious

Art Unit: 2193

differences which would have been obvious to one of ordinary skill in the art of program development at the time of invention such as simply (i) omitting/adding steps or elements along with their functions, and/or (ii) implementing the method steps with means for performing the steps, and/or (iii) system, device, medium.

The following example is given:

Per claim 50:

Patent '848 claim 15 recites: receive a data representation language representation of a first computer programming language object and generate the first object from the data representation language representation of a first object, wherein the first object is an instance of a class in the computer programming language ("A method for representing a state of a process in a data representation language in a distributed computing environment...executing the process within a first device; converting a current computation state of the process into a data representation language representation of the current computation state, wherein the computation state of the process comprises information about the execution state of the process within the first device...wherein the data representation language representation of the current computation state of the process is configured for use in reconstituting the process and resuming execution of the process") as recited in instant claim 50. The instant claim 50 does not explicitly use same words and steps such as "storing the data representation...of the process" and "resuming execution of the process." However, it would have been obvious for one of ordinary skill in the pertinent art at the time the instant invention was made to modify the '848 patent claim by simply omitting the recited steps for the purpose of expediting the method.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1, 2, 4, 6-41, 43-71, and 73-90 are rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson ("XML JavaBeans Integration, Part 3," 7/1999) in view of Allen (US Patent 6,658,625).

Per claim 1:

Johnson discloses:

-a process executing within a virtual machine providing a first computer programming language object to a compilation process of the virtual machine, wherein the first object is an instance of a class in the computer programming language (i.e. XMLBeans can transform a JavaBean in memory into an XML document," page 1 first paragraph, lines 1-3)

- the compilation process of the virtual machine converting the first object into a data representation language representation of the first object (i.e. "read and write JavaBeans objects as XML documents," page 1 paragraph 7, line 3); wherein the data representation language representation of the first object is configured for use in generating a copy of the first

Art Unit: 2193

object (i.e. can transform an XML document...into a running JavaBean,” page 1 first paragraph, lines 2-3).

Johnson does not explicitly teach processing the first object into an intermediary table representation of the first object, wherein at least one entry of the intermediary hash table and processing the intermediary table representation of the first object into the data representation language representation of the first object. Although Johnson does not state if the properties of the JavaBean instances are represented in a hash table for conversion between JavaBean and XML, Allen teaches using such hash table was known in the pertinent art, at the time applicant's invention was made, to enhance fast data access and save memory resources (i.e. col. 12 lines 13-19). It would have been obvious for one having ordinary skill in the art to modify Johnson's disclosed system to parse into a hash table to store and retrieve the instance properties. The modification would be obvious because one having ordinary skill in the art would be motivated to achieve fast data access in the conversion process between JavaBean and XML.

Allen further discloses the hash table representation including a hash key including a name of an instance variable of the first object and a value for the instance variable (i.e. col. 13 lines 33-44).

Per claim 2:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- the compilation process converting the one or more objects into data representation language representations of the one or more objects (i.e. page 5 paragraphs 3-5).

Art Unit: 2193

Per claim 4:

The rejection of claim 1 is incorporated, and further, Allen teaches:

- for each of the one or more instance variables in the first object, generating an entry in the intermediary table representation of the first object, wherein the entry for each of the one or more instance variables includes an identifier of the instance variable and a value of the instance variable (i.e. col. 13 lines 33-39).

Per claim 6:

The rejection of claim 4 is incorporated, and further, Allen teaches:

- for each of one or more entries in the intermediary table representation of the first object, generating a corresponding element in the data representation language representation of the first object, wherein the element in the data representation language representation of the first object includes an identifier of the instance variable and a value of the instance variable (i.e. col. 13 lines 39-44).

Per claim 7:

The rejection of claim 6 is incorporated, and further, Allen teaches:

the one or more elements in the data representation language representation of the first object are configured for use in initializing one or more corresponding instance variables in the copy of the first object (i.e. col. 13 lines 45-47).

Per claim 8:

Art Unit: 2193

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- providing an application programming interface (API) for the compilation process, wherein the API comprises interfaces to one or more methods of the compilation process configured for use by processes executing within the virtual machine to convert computer programming language objects into data representation language representations of the objects (i.e. page 7, last paragraph, lines 1-4).

Per claim 9:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

-said data representation language is extensible Markup Language (XML) (i.e. "XML document," page 1, lines 2-3).

Per claim 10:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- said computer programming language is the Java programming language (i.e. "a JavaBean," page 1, lines 2-3).

Per claim 11:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

-the virtual machine is a Java Virtual Machine (JVM) (i.e. "running JavaBean," page 1, line 3).

Art Unit: 2193

Per claim 12:

Johnson teaches:

-a virtual machine receiving a data representation language representation of a first computer programming language object from a first process a decompilation process of the virtual machine generating the first object from the data representation language representation of the first object, wherein the first object is an instance of a class in the computer programming language (i.e. XMLBeans can transform a JavaBean in memory into an XML document,” page 1 first paragraph, line 2; “read and write JavaBeans objects as XML documents,” page 1 paragraph 7, line 3; can transform an XML document...into a running JavaBean,” page 1 first paragraph, lines 1-3).

Johnson does not explicitly teach and the decompilation process of the virtual machine providing the first object to a second process executing within the virtual machine. However, Allen teaches it was known in the pertinent art, at the time applicant's invention was made, to reconstitute the object state in a distributed system (i.e. col. 3 lines 12-24). It would have been obvious for one having ordinary skill in the art to modify Johnson's disclosed system to incorporate the teachings of Allen. The modification would be obvious because one having ordinary skill in the art would be motivated to reproduce object state in a distributed system.

Per claim 13:

The rejection of claim 12 is incorporated, and further, Johnson teaches:

Art Unit: 2193

- the first object references one or more computer programming language objects, wherein the representation of the first object includes representations of the one or more referenced objects (i.e. page 5 paragraph 3-4).

Per claim 14:

The rejection of claim 13 is incorporated, and further, Johnson teaches:

- the decompilation process generating the one or more referenced objects from the representations of the one or more referenced objects included in the representation of the first object (i.e. page 5 paragraph 5).

Per claim 15:

Allen further discloses: processing the data representation language representation of the first object into an intermediary table representation of the first object; and generating the first object from the intermediary table representation of the first object (i.e. col. 13 lines 33-39).

Per claim 16:

The rejection of claim 15 is incorporated, and further, Allen teaches:

- one or more elements each representing an instance variable of the first object, wherein each element in the data representation language representation comprises an identifier for the instance variable represented by the element and a value for the instance variable represented by the element (i.e. col. 13 lines 39-44).

Per claim 17:

The rejection of claim 16 is incorporated, and further, Allen teaches:

Art Unit: 2193

-representation language representation of the representation of the first object comprises generating an entry in the intermediary table representation of the first object for each of the one or more elements in the data representation language representation of the first object (i.e. col. 13 lines 45-55).

Per claim 18:

The rejection of claim 17 is incorporated, and further, Allen teaches:

- instantiating the first object as an instance of the class; and for each of the one or more entries in the intermediary table representation of the first object, initializing a corresponding instance variable in the first object in accordance with the entry (i.e. col. 13 lines 45-47).

Per claim 19:

The rejection of claim 17 is incorporated, and further, Johnson teaches:

-instantiating the first object as an instance of the class; and for each of the one or more entries in the intermediary table representation of the first object, invoking a method corresponding to the identifier of the instance variable from the entry to initialize a corresponding instance variable in the first object to the value of the instance variable from the entry (i.e. page 13, lines 2-3).

Per claim 20:

The rejection of claim 12 is incorporated, and further, Johnson teaches:

-the data representation language representation of the first object comprises an identifier of the class of the first object, and wherein the decompilation process generating the first object from the data representation language representation of the first object comprises instantiating

Art Unit: 2193

the first object as an instance of the class associated with the class identifier (i.e. page 5 paragraph 3-5).

Per claim 21:

The rejection of claim 16 is incorporated, and further, Johnson teaches:

-providing an application programming interface (API) for the decompilation process, wherein the API comprises interfaces to one or more methods of the decompilation process configured for use by processes executing within the virtual machine to generate computer programming language objects from data representation language representations of the objects. (i.e. page 7 last paragraph, lines 1-4).

Per claim 22-24: they are another method versions of claims 9-11, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 9-11 above.

Per claim 25:

Johnson teaches:

-a first virtual machine receiving from a first process a computer programming language object, wherein the object is an instance of a class in the computer programming language the first virtual machine generating a representation of the object in a data representation language subsequent to said receiving generating the data representation language representation of the object (i.e. XMLBeans can transform a JavaBean in memory into an XML document,” page 1 first paragraph; “read and write JavaBeans objects as XML documents,”

Art Unit: 2193

page 1 paragraph 7; “ can transform an XML document...into a running JavaBean,” page 1 first paragraph).

Johnson does not explicitly teach generating a message in the data representation language and sending the message to a second process. However, Allen teaches generating XML message of an object and providing such XML representation to another process was known in the pertinent art, at the time applicant's invention was made, to reconstitute the object state from the decompilation in a distributed system (i.e. col. 3 lines 12-24). It would have been obvious for one having ordinary skill in the art to modify Johnson's disclosed system to incorporate the teachings of Allen. The modification would be obvious because one having ordinary skill in the art would be motivated to reproduce object state in a distributed system.

Johnson in view of Allen further teaches: the second process generating a copy of the computer programming language object from the data representation language representation of the object included in the message (i.e. page 1 first paragraph).

Per claim 26:

The rejection of claim 25 is incorporated, and further, Johnson teaches:

-the object references one or more computer programming language objects, and wherein said generating a representation of the object in a data representation language comprises generating data representation language representations of the one or more objects(i.e. page 5 paragraph 3-5).

Per claim 27:

Art Unit: 2193

The rejection of claim 25 is incorporated, and further, Johnson teaches:

-for each of the one or more instance variables in the object, generating an element in the data representation language representation of the first object, wherein the element for each of the one or more instance variables includes an identifier of the instance variable and a value of the instance variable (i.e. page 7 last paragraph, lines 1-4).

Per claim 28:

The rejection of claim 25 is incorporated, and further, Johnson teaches:

-the second process receiving the message including the data representation language representation of the object; the second process providing the data representation language representation of the object to a second virtual machine; the second virtual machine generating the copy of the object from the data representation language representation of the object; and the second virtual machine providing the copy of the object to the second process (i.e. page 5 paragraph 3-4).

Per claim 29:

The rejection of claim 28 is incorporated, and further, Johnson teaches:

-the first object references one or more computer programming language objects, wherein the data representation language representation of the first object includes data representation language representations of the one or more referenced objects, and wherein said generating the copy of the object from the data representation language representation of the object comprises generating copies of the one or more referenced objects from the data representation language representations of the one or more referenced objects (i.e. page 5 paragraph 5).

Art Unit: 2193

Per claim 30:

The rejection of claim 28 is incorporated, and further, Johnson teaches:

-instantiating the copy of the object as an instance of the class; and
for each of the one or more elements in the data representation language
representation of the object, initializing a corresponding instance variable
in the copy of the object in accordance with the element (i.e. page 7 last paragraph lines 1-4).

Per claims 31-33, they are another method versions of claims 22-24, respectively, and are
rejected for the same reasons set forth in connection with the rejection of claims 22-24 above.

Per claim 34-39: they are another method versions of claims 25-33, respectively, and are
rejected for the same reasons set forth in connection with the rejection of claims 25-33 above.

Per claims 40, 41, and 43-49, they are the device versions of claims 1, 2, 4 and 6-11,
respectively, and are rejected for the same reasons set forth in connection with the rejection of
claims 1, 2, 4 and 6-11 above.

Per claim 50:

Johnson discloses:

-receive a data representation language representation of a first computer programming language object (i.e. XMLBeans can transform a JavaBean in memory into an XML document,” page 1 first paragraph)

- generate the first object from the data representation language representation of a first object, wherein the first object is an instance of a class in the computer programming language (i.e. “read and write JavaBeans objects as XML documents,” page 1 paragraph 7; “can transform an XML document...into a running JavaBean,” page 1 first paragraph).

Johnson does not explicitly teach a system comprising a processor and memory to execute the instructions. Allen teaches it was known in the pertinent art, at the time applicant's invention was made, to reconstitute the object state in a distributed system (i.e. col. 3 lines 12-24). It would have been obvious for one having ordinary skill in the art to modify Johnson's disclosed system to incorporate the teachings of Allen. The modification would be obvious because one having ordinary skill in the art would be motivated to execute and reproduce object state in a distributed system.

Per claims 51-61, they are another device versions of claims 12-24 respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 12-24 above.

Per claims 62-70, they are the medium versions of claims 25-33, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 25-33 above.

Per claims 71 and 73-77, they are another medium versions of claims 1, 2, 4 and 6-11, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1, 2, 4 and 6-11 above.

Art Unit: 2193

Per claims 78-83, they are the medium versions of claims 12-24, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 12-24 above.

Per claims 84-90, they are the medium versions of claims 25-33, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 25-33 above.

7. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson ("XML JavaBeans Integration, Part 3," 7/1999), in view of Allen (US Patent 6,658,625), and further in view of Gillam ("Java Liaison" column, 3/1999).

Per claim 5:

Johnson and Allen do not explicitly teach that the first object comprises a plurality of instance variables with the same identifier, and wherein the entry for each of the plurality of instance variables with the same identifier further includes an enumeration value that uniquely identifies the instance variable in the plurality of instance variables with the same identifier. However, Gillam teaches using enumeration was well known in the pertinent art, at the time applicant's invention was made, to specify individually legal values for a particular type (i.e. see Enumerated types example in page 2). It would have been obvious for one having ordinary skill in the art to modify Johnson and Allen's disclosed system to incorporate the teachings of Gillam. The modification would be obvious because one having ordinary skill in the art would be motivated to enumerate all of the possible values for a variable of the type for run-time efficiency (i.e. see Enumerated types example in page 2).

Art Unit: 2193

Response to Arguments

8. Applicant's arguments filed 4/23/2007 have been fully considered but are moot in view of the new ground(s) of rejection.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Insun Kang whose telephone number is 571-272-3724. The examiner can normally be reached on M-R 6:30-5 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, MENG AI AN can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

IK
AU 2193


MENG-AI AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100